DOCUMENTAZIONE PROGETTO “IL DUNGEON OSCURO”

Struttura Progetto

-main.c

-giocatore.h

-inventario.h

-inventario.c

-negozio.c

-negozio.h

-salvataggi.h

-salvataggi.c

-systemclear.h

-systemclear.c

-tabellemissioni.h

-tabellemissioni.c

-nuovoGioco.h

-nuovoGioco.c

-menuvillaggio.h

-menuvillaggio.c

-menuMissioni.h

-menuMissioni.c

-missione1espdun.h

-missione1espdun.c

-missione2espdun.h

-missione2espdun.c

-missione3espdun.h

-missione3espdun.c

-missionefinale.h

-missionefinale.c

Spiegazione struttura progetto

- Il progetto è stato diviso in vari file e per i quali c'è un **file header** correlato che definisce le funzioni principali del file.
- La **struttura** principale è ''giocatore.h'', la quale definisce i dati base come vita, monete, attacco e dei boolean utili per tracciare i progressi durante lo svolgimento del gioco (es: ha_spada, missione_palude_completata). Perciò passando un unico puntatore alle funzioni (quello di giocatore) riusciamo a garantire la portabilità delle modifiche in tutto il gioco.
- Le varie **funzioni** necessarie al funzionamento del gioco sono divise ognuna in un file dedicato. Per esempio, il negozio ha un suo file e ogni missione ha un file dedicato (es: ''missione1espdun'').
- I **salvataggi** avvengono mediante l'utilizzo di file di testo, salvando quindi lo stato completo del giocatore in dei file contenuti nella cartella ''salvataggi''. Utilizzando questa struttura:

❖ **data e ora**
❖ **vita -- monete -- attacco**
❖ **ha_spada -- ha_armatura -- ha_spada_eroe -- ha_eroe**
❖ **missione_palude_completata -- missione_magione_completa -- missione_grotta_completata**

- Per garantire la **compatibilità** sia su sistemi Windows che su sistemi Unix/Linux abbiamo utilizzato il preprocessore #ifdef _WIN32. Questo ci ha permesso di mettere un filtro su quali librerie includere in base a quale sistema operativo sta compilando il codice, cosí che il codice sia adattivo. Per esempio, per Windows esiste una libreria specifica per la gestione delle directory (<direct.h>)
- Nella **gestione dell'input**, invece di usare solo scanf, che spesso lascia residui nel buffer di input, in diverse sezioni del gioco (come nel caricamento dei salvataggi) abbiamo preferito l'uso di fgets combinato con il parsing delle stringhe, per una lettura migliore.

Organizzazione del lavoro

Il lavoro è stato suddiviso equamente tra i componenti del gruppo. Secondo questa divisione:

- **[Elia Cavasin]:** Gestione menu Iniziale, input/output, Input da tastiera, output di stringhe
- **[Lorenzo Galloni]:** Struttura Giocatore, funzioni necessarie, input/output, restituzione della struttura aggiornata o elenco salvataggi
- **[Lorenzo Giudice]:** Creazione menu e funzionamento tre missioni, generazione Dungueon
- **[Denis Stetco]:** Creazione menu e funzionamento missione finale

Principali difficoltà incontrate

1. **Manipolazione dei puntatori:** Una delle difficoltà riscontrate durante la creazione del progetto è stata la gestione dei puntatori. In particolare, quella del giocatore che richiedeva un'elevata attenzione nella distinzione tra la struttura del giocatore definita nel codice e le istanze effettive di esso presenti in memoria. Le difficolta sono poi aumentate quando più funzioni modificano la stessa struttura.
2. **Gestione dei file:** Un'altra difficoltà che è stata riscontrata da tutti e quattro i componenti del gruppo è stata la suddivisione tra file header e i file d'implementazione. Mappare correttamente le dichiarazioni delle funzioni nei .h e le loro implementazioni in .c ha richiesto una elevata attenzione per evitare che ci fossero errori di dichiarazioni o linker.
3. **Integrazione dei moduli:** L'integrazione dei diversi moduli sviluppati separatamente è stato un punto delicato. Infatti, ogni parte del progetto funzionava in modo corretto in maniera isolata, ma l'unione ad esempio del menu, le missioni, l'inventario ed i salvataggi hanno portato delle dipendenze che non erano immediatamente evidenti. Tutto ciò ha richiesto una fase di test approfondita per garantire che le modifiche di un modulo non compromettesse il funzionamento degli altri.
4. **Coordinamento del gruppo:** Il coordinamento del gruppo ha rappresentato più una sfida organizzativa che tecnica. Lavorando su moduli diversi dello stesso progetto è stato necessario una costante comunicazione tra i membri per garantire che le parti fossero compatibili tra loro. Il continuo confronto ha permesso di poter individuare in maniera più rapida soluzioni a problemi e di risolverli in modo collaborativo. Tutto questo ha sicuramente contribuito a migliorare l'organizzazione del progetto e rafforzare la capacità di lavorare in gruppo.